2018

An Explorative Study Into Situational Artefact Construction in Business Rules Management

Sam Leewis
HU University of Applied Sciences Utrecht, sam.leewis@hu.nl

Koen Smit
HU University of Applied Sciences Utrecht, koen.smit@hu.nl

Martijn Zoet
Zuyd University of Applied Sciences, martijn.zoet@zuyd.nl

31ST BLED ECONFERENCE: DIGITAL TRANSFORMATION: MEETING THE CHALLENGES
JUNE 17 - 20, 2018, BLED, SLOVENIA, CONFERENCE PROCEEDINGS
A. Pucihar, M. Kljajić Borštnar. P. Ravesteijn, J. Seitz & R. Bons

An Explorative Study Into Situational Artefact Construction in Business Rules Management

SAM LEEWIS, KOEN SMIT & MARTIJN ZOET

Abstract The implementation of software products is a time-consuming activity and requires specific expertise to be completed successfully. This is especially the case in research fields where there is no or little tool support available, such as Business Rules Management (BRM) and Business Rules Management Solutions (BRMS). Tool support is essential to successfully guide the organizational implementation of a BRMS. Motivated by the diversity of organizational structures and their BRMS implementation contexts, we design a situational-aware framework for the organizational implementation of BRMS. The framework is based on the theory of situational artefact construction. Using situational artefact construction, we study 13 BRMS implementation cases distributed over the financial and public sectors in the Netherlands. Based on the results of the cases analysed we present a framework with three main artefacts that are a stepping-stone towards further research on situational implementation methodology in the BRM field.

Keywords: • Business Rules Management • Business Rules Management Solution • Situational Artefact Construction •

CORRESPONDENCE ADDRESS: Sam Leewis, Digital Smart Services, HU University of Applied Sciences Utrecht, Nijenoord 1, 3552 AS Utrecht, The Netherlands, e-mail: sam.leewis@hu.nl. Koen Smit, Digital Smart Services, HU University of Applied Sciences Utrecht, Nijenoord 1, 3552 AS Utrecht, The Netherlands, e-mail: koen.smit@hu.nl. Martijn Zoet, Optimizing Knowledge-Intensive Business Processes, Zuyd University of Applied Sciencess, Ligne 1, 6131 MT Sittard, The Netherlands, e-mail: martijn.zoet@zuyd.nl

DOI https://doi.org/10.18690/978-961-286-170-4.12 ISBN 978-961-286-170-4

Available at: http://press.um.si.

1 Introduction

An increasing amount of laws and regulations and the demand for IT automation raises the need for handling business rules in a proper way (Boyer & Mili, 2011; Graham, 2007). This occurs since the introduction of the separation of concerns principle (van der Aalst, 1998) and Business Rules Management (BRM) (Boyer & Mili, 2011; Morgan, 2002; Ross, 2013). The fact that organizations increasingly consider business rules as a separate 'concern' to manage in combination with the growing amount of business rules results in organizations seeking a solution to manage their business rules. We refer to such a solution as a Business Rules Management Solution (BRMS). A BRMS can be implemented to manage the elicitated, designed, specified, and deployed business rules (Nelson, Rariden, Sen, & Texas, 2008; Smit & Zoet, 2016; Taylor, 2011; Zoet & Versendaal, 2013)

In the field of information systems, the domain of Business Rules Management (BRM) is a relatively young subject of study and gained the interest from researchers the past several years (Zoet, 2014). This is especially the case for the organizational implementation of a BRMS (Nelson, Peterson, Rariden, & Sen, 2010).

This study focuses on the latest one motivated by the significant difference in scientific contributions for organizational implementations support of BRMS compared to technical implementations support (Nelson et al., 2010; Zoet, 2014). Therefore, the researchers intend to answer the following research question: "*How to develop a framework that supports the organisational implementation of a business rules management solution?*''

In this research, we adopt the situational artefact construction technique (Hevner, March, Park, & Ram, 2004; Winter, 2011a, 2011b) to answer this research question and design a framework. The framework involves three main artefacts 1) the BRMS analysis tool to extract the building blocks of a BRMS; 2) the BRMS construction process, which analyses the identified design problems and problem classes in order to elicit design situations and design factors; and 3) the BRMS metamodel to facilitate the instantiation of the elicited data from the BRMS analysis tool and BRMS construction process.

Situational artefact construction is applied in other research fields such as software product management (Bekkers, van de Weerd, Brinkkemper, & Mahieu, 2008; van de Weerd, 2009), and business process management (Bucher & Winter, 2010; Ravesteyn & Jansen, 2009). The BRM research field lacks this type of research and therefore this research is of an explorative nature. In this study, the developed BRMS framework is validated with 13 BRMS implementation cases from the financial and the public industries in the Netherlands. Additional validation of framework feasibility is conducted by analysing the elicited data, specifying design situations and design factors, and instantiating the BRMS metamodel. Therefore, the main contribution of this paper is of twofold; 1) the BRMS implementation framework is described, and 2) the BRMS implementation framework is applied to the 13 BRMS implementation cases.

The remainder of the paper is structured as follows: First, the BRMS in the context of the related work is discussed. Second, this is followed by the research methods utilized for this study. Next, the BRMS implementation framework is proposed as a proof of concept together with its major elements. Subsequently, the BRMS implementation framework is validated through expert interviews. Lastly, the conclusions are provided that can be drawn from the results, together with a critical view of the results of this study followed by possible future research.

2 Related Work

Business rules describe the state of affairs of what the business demands (Business Rules Group, 2003; Morgan, 2002). For this research, we adopt the definition of a business rule by Morgan (2002): ‘*’a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business.*’’

To improve the grip on business rules, organizations search for a controlled approach to support the management of business rules, this approach is called BRM (Boyer & Mili, 2011; Ross, 2003). BRM is defined as: ‘*’a systematic and controlled approach to get a grip on business decisions and business logic to support the elicitation, design specification, verification, validation, deployment, execution, governance, and monitoring of both business decisions and business logic*’’ (Smit, Zoet, & Berkhout, 2017). The BRM capabilities are part of an approach or method with the goal to translate sources like law and regulations or internal policies into products or services. Smit and Zoet (2016) explains in their work, in detail, the specifics of the BRM capabilities. Earlier mentioned capabilities of BRM need to be supported by some sort of IT solution. A specific IT solution in the context of BRM is a BRMS (Graham, 2007).

A BRMS is ‘’a configuration of capabilities which supports the Elicitation, Design, Specification, Verification, Validation, Deployment, Execution, Monitoring, and Governance of business rules’’ (Leewis, Smit, & Zoet, 2018). Each implementation of a BRMS is one capability or a combination of the nine Capabilities (as shown in Figure 1) which an organization can configure for their own purposes to create, implement, and manage business rules (Smit & Zoet, 2016; Zoet, 2014; Zoet & Versendaal, 2013). The technical implementation of a BRMS is covered extensively in researchArnott & Pervan, 2014; Rosca & Wild, 2002; Zoet, 2014). Analysis of the literature on organizational implementation of a BRMS results in no relevant related work (Nelson et al., 2008; Zoet, 2014). Previously conducted research has shown that different BRMSs have a common Design Problem (Aier, Riege, & Winter, 2008; Baumöl, 2005; Bucher & Winter, 2010; Klesse & Winter, 2007). A common Design Problem is the difference between the goal state and the current state of a system, a BRMS, and is an indication that common Problem Classes, for which Design Solution can be created, exists (Winter, 2011b). Winter, (2011b) depicts a Problem Class as a set of comparable Design Problems. A Problem Space is a collection of multiple Problem Classes. An instantiation of a Problem Class in a specific organization is defined as a Design Solution. Specific for the BRMS Problem

Space, the Design Solution is a specific configuration of the earlier mentioned nine BRMS Capabilities (Smit & Zoet, 2016; Zoet & Versendaal, 2013). Every configuration of BRMS Capabilities is influenced by certain factors, so-called Situational Factors. Situational Factors describe the context in which an information system artefact or organization has to operate in a way that the deployed artefact fits the context of the environment.

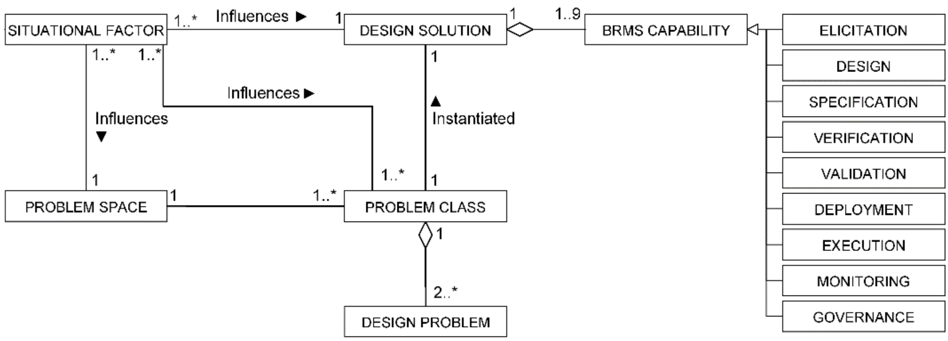

Figure 1: BRMS problem space

Research identifying situational factors is conducted in the situational method engineering research field (Brinkkemper, 1996; Karlsson, Ågerfalk, & Hjalmarsson, 2001; Rolland & Prakash, 1996; van Slooten & Hodes, 1996), with specific applications in software product management (Bekkers et al., 2008; van de Weerd, 2009), and business process management (Bucher & Winter, 2010; Ravesteyn & Jansen, 2009).

The BRMS implementation framework can be constructed as a situational artefact, which ensures that the framework could be adapted to solve different design problems within a problem class (Winter, 2011b). Winter (2011b) proposed in his work a technique to create situational artefacts, which is used as a guideline in this study.

3 Research Method

The Situational artefact construction technique from Winter (2011a, 2011b) is used as a research method to create a situational artefact (the BRMS implementation framework) to support BRMS implementations. To structure and ground the research activities, Hevner's Design Science Research is used (Hevner et al., 2004). The situational artefact construction technique exists of eleven steps and is adapted specifically on situational artefact construction in the BRMS domain, adaptation is needed because the focus of the initial situational artefact construction is on a generic level in the context of software development. In the BRM research field, a more qualitative approach is needed because the field lacks standards as a base to perform quantitative analysis on. The original process is focused on creating a situational artefact for any given research field. This technique embodies the following steps: 1) Initial demarcation of the design problem class, 2)

Identification of potential contingency factors, 3) Field study analysis of design problems in practice, 4) Refining specifications of the design problem class, 5) Calculation of ultra-metric distances, 6) Determination of a useful level of generality, 7) Specification of design situations, 8) Identifying characterizing design factors, 9) Linking design factors to related design problems, 10) Deriving elementary problem-solving actions, and 11) Deriving method configuration rules (Winter, 2011a, 2011b). The eleven situational artefact construction steps are elaborated further in the paper with data used from a running experiment (Leewis, Smit, & Zoet, 2018; Leewis, Smit, Zoet, & Berkhout, 2018). The 13 gathered BRMS implementation cases were used as input for creating the BRMS implementation framework and are explained in detail in the work of Leewis, Smit, Zoet, et al. (2018).

4 BRMS Implementation Framework

This section describes the elements of the developed BRMS implementation framework, which are all focussed on supporting organizational implementations of a BRMS. The first element of the BRMS implementation framework is the BRMS analysis tool. This tool is mainly focussed to gather BRMS implementation cases and is explained in detail in the work of Leewis, Smit, & Zoet (2018). The second element of the BRMS implementation framework is the BRMS construction process. This process aims at analysing data from the BRMS analysis tool and to use this data for the design of the BRMS metamodel. Hence, the third element of the BRMS implementation framework is the BRMS metamodel where all the elements important for a BRMS implementation are specified. The situational artefact construction method from Winter (Winter, 2011a, 2011b) is used as a guideline to create a situational artefact in the BRM domain.

4.1 Initial demarcation of the design problem class

Discovery of the BRM problem space is needed to identify the existing knowledge on creating a situational artefact in the field of BRM. The goal of this step is to discover concepts in the field of BRM to support the creation of a situational artefact. A literature review is conducted to create an overview of the existing body of knowledge. This step is covered extensively in the work of Leewis, Smit, & Zoet (2018).

4.2 Identification of potential contingency factors

The literature review is also used for the identification of potential contingency factors. This literature review resulted in the creation of the BRMS analysis tool which purpose is to gather and analyse BRMS implementation cases (Leewis, Smit, & Zoet, 2018). Additional interviews are conducted with members of the BRM community to validate the discovered problem classes, and the contingency factors included in BRMS analysis tool. The BRMS analysis tool is constructed by combining knowledge derived from literature and knowledge derived from interviews with BRM experts.

4.3 Field study analysis of design problems in practice

The goal of this field study is to collect data on different BRMS implementations. The set of implementations used in the work of Leewis, Smit, Zoet, et al. (2018) are used in this study to create the BRMS implementation framework. These collections of different implementations can create an overview of clusters of situational factors that influence the different problem classes in the business rules management solution problem space. Thereby, creating a situational artefact for each situation. To reduce a large number of contingency factors into a relevant set of design factors, a qualitative approach of a Principal Component Analysis (PCA) was performed. The main goal of a PCA is reducing a list of potential contingency factors to relevant design factors (Abdi & Williams, 2010; Jolliffe, 2002).

4.4 Refining specifications of the design problem class

The previous three steps focus on creating a relevant list of design factors. This step focusses on specifying and refining the design factors into more specific and refined design problem classes. The design problem classes identified in earlier steps should be refined more to ensure define the degree of homogeneity of the solutions. This will result in the excluding of ‘’outlier’’ design solutions and thereby ensuring the degree of homogeneity. The problem classes identified are: 1) Elicitation, 2) Design, 3) Specification, 4) Verification, 5) Validation, 6) Deployment, 7) Execution, 8) Monitoring, and 9) Governance. The problem classes are identical with the BRM capabilities (Smit & Zoet, 2016). The next step will cluster the solutions into relevant clusters which are representative towards all the gathered BRMS implementation cases.

4.5 Calculation of ultra-metric distances

Problem classes can be divided into a number of generic design situations depending on the degree of generality. The generic design situations are the specified solutions depending on the number of clusters selected in the problem class. Based on the Euclidian distance, the similarity (or dissimilarity) of two design solutions within a problem class can be represented by an ultrametric-distance (Deza & Deza, 2016). The cases and their distances are visualized using a tree-like graph. Ultrametric-distances can be visualized by a graph whose Y-axis represents generality and whose X-axis represents the set of observed cases. The similarity or dissimilarity of two design solutions (or more) corresponds to the generality level of their relation. If the similarity is high, their relation is represented on a lower level of generality (Winter, 2011a). Figure 2 shows the ultra-metric distance of the elicitation problem class (because of space constraints only the elicitation capability is included in this paper).

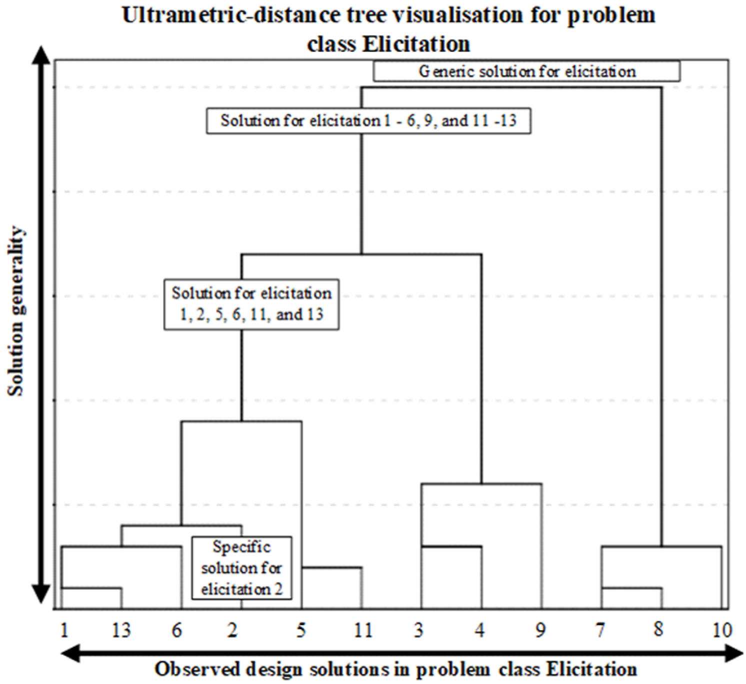

Figure 2: Ultrametric-distance visualization

This graph contains 12 solutions (elicitation 1-11, and 13) for the elicitation problem class. Important to mention is that the sample includes 13 implementations, only case 12 didn't implement the elicitation problems class. Therefore, case 12 is not included in the analysis of the problem class. Observed case 2 is represented by its own specific solution, Elicitation solution 2. This level of generality is the same for the other observed cases on this level which are represented by their own specific solution. Observed case 1, 2, 5, 6, 11 and, 13 are represented by a more generic solution, Elicitation solution 1, 2, 5, 6, 11 and, 13. The observed cases 1 - 6, 9, and 11, 13 are represented by an even more generic solution, Elicitation solution 1 - 6, 9, and 11, 13. At the top level, the generic solution contains all the observed cases and is the most generic representation towards all the observed cases.

4.6 Determination of level of generality

This step will focus on what level of generality is needed to have optimal cluster consistency in a design solution. In order to not only visualize the generic solutions (as shown in Figure 2), but also specify the generic solutions (as shown in Figure 3), k-means cluster analyses are applied to the BRMS implementation cases. Thereby, determining the optimal number of clusters for the design solution. An optimal number of clusters is where the number of clusters has the lowest error sum, as shown in Figure 3. The k-mean cluster analyses were conducted with the minimum amount of cluster and the maximum amount of cluster possible in the problem class. These cluster analyses were conducted for each problem class because the possibility exists that a solution contains only one capability (problem class). For each solution the error total which is calculated from the distances of all implementation cases to the centre of their clusters. Based on this total, the so-called elbow criterion is used (Everitt, Landau, Leese, & Stahl, 2011; Hardle & Simar, 2007; Winter, 2011b). The elbow criterion indicates which increased number of clusters leads to an above-average improvement in the error sum. The error sum is plotted on the y-axis, and the number of clusters is plotted on the y-axis, an elbow arises for the 4-cluster, 7-cluster and 10-cluster solution as shown in Figure 3. The 4-cluster solution is selected due to the lowest error sum compared to the 7- and 10-cluster solutions.

Figure 3: Elicitation problem class elbow criterion

4.7 Specification of design situations

The design situations need more than only a formal definition (which is done by the ultrametric-distance calculation) but also need semantic interpretation (i.e. by specifying the design problem types). The ultrametric-distances are used, for semantically specifying the design situations. The problem classes are specified into their preferred cluster consistency. The Elicitation problem class is specified into a 4-cluster solution, and these four different clusters are specified using a mean comparison analysis. This analysis aims at specifying the design situations. The goal of this is to create different design situations, as shown, for example, in Table 1. Thereby, aiming at what is the exact reason why these design solutions are created into a cluster. Elements of all the design situations in a problem class which were identical were excluded from the design situation specification. Specifying design situations aims at showing factors which differentiate a design situation from another design situation, this will not help when identical elements do not differentiate a design situation from another design situation. These factors need to be characterized further.

Table 1: Elicitation design situations

Design Situation 1:	**Design Situation 2:**
Public sector focussed	Public sector focussed
Organisations with 2001 – 5000 employees	Organisations with more than 5000 employees
Organisation implementation focussed	Line of business implementation focussed
Fully manual elicitation focussed (autonomy level 1)	Capability is supported with a complete set of action alternatives (autonomy level 2)
Database data is used for elicitating business rules	No database data is used for elicitating business rules
Outcome of the capability is not a relevant set of selected sources	Outcome of the capability is a relevant set of selected sources
Scenario analysis is conducted	No scenario analysis is conducted
Design Situation 3:	**Design Situation 4:**
Financial sector focussed	Financial sector focussed
Organisations with more than 5000 employees	Organisations with 251 -500 employees
Line of business implementation focussed	Line of business implementation focussed
Capability is supported with a narrowed down set of action alternatives (autonomy level 3)	Fully manual elicitation focussed (autonomy level 1)
No database data is used for elicitating business rules	No database data is used for elicitating business rules
Outcome of the capability is not a relevant set of selected sources	Outcome of the capability is a relevant set of selected sources
No scenario analysis is conducted	Scenario analysis is conducted

4.8 Identifying characterizing design factors

Each design situation consists of characterizing design factors and needs to be specified further. Every design factor has different values, these values influence the characterization of the design factor and thereby the design situation. An example of a design factor: Capability leader, which characterizes the department that is in the lead regarding the specific capability, shown in Table 2. The value of this capability could be one of the following departments: IT, the business, or a central IT/business group. Characterizing design factors are defined together with their values and related problem classes. Characterizing design factors define the design situations, but there are also design factors which do not characterize design situations but are still important for solving design problems. Non-characterizing design factors (ID# 42 and 43 in Table 2) are not specifically solving design problems but are directly related to the problem classes, which in their turn solve certain design problems. The reason that these design factors are non-characterizing is that for all the design situations in a problem class these design factors have the same value.

Table 2: Examples (non)-characterizing design factors

ID:	Problem class:	Design factor:	Value:	Description:
4	Design, Specification, Deployment, Execution, Monitoring, and Governance	Capability leader	IT, Business, or Central IT/Business group	The Capability leader design factor characterizes which department is in the lead of the specific capability
5	All	Capability autonomy	Autonomy level 1 - 10	The capability autonomy characterises the level of autonomy on which the capability in the BRMS operates
6	Elicitation	Elicitation source (Data)	Yes/No	Database data is used as a source for elicitation.
7	Elicitation	Relevant set of selected sources	Yes/No	The output of elicitation is a relevant set of selected sources.
8	Elicitation	Scenario analysis	Yes/No	Scenario analysis is used during elicitation.
42	Elicitation	Elicitation source (People)	Yes	People are used as a source for elicitation.
43	Elicitation	Elicitation source (Documents)	Yes	Documents are used as a source for elicitation.

4.9 Linking design factors to related design problems

The characterized design factors described in the earlier sections need to be linked to the 24 proposed design problems which are discovered in the work of (Leewis, Smit, Zoet, et al., 2018) and adopted for this study. All earlier conducted steps analyse the existing design solutions. These design solutions are cases of successful BRMS implementations. These design solutions were created with a certain purpose, solving an existing problem, and therefore the design factors can be qualitatively interpreted and linked to the known design problems, as shown in Table 3. The 24 known design problems were mapped against the identified and characterized design factors. An example of this is the design problem ''Increase Elicitation productivity'' which could be positively impacted by letting the system take over some tasks. The design factors solving this design problem are: (capability autonomy), using data as a source when eliciting, the output of elicitation is a relevant set of selected sources (which could be reused), performing a scenario analysis which is based on business scenarios. In short, a certain combination of characterized design factors could solve a certain design problem.

Table 3: Examples Design factors linked to Design Problems

	Design factors:																								
Design problems:	**5**	**6**	**7**	**8**	**9**	**10**	**14**	**16**	**21**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**
Increase Elicitation productivity	X	X	X	X			X																		
Construct library of decisions			X	X	X	X			X	X		X				X								X	
Ensure artifact relationship insight								X			X	X	X	X	X	X	X	X	X	X	X	X	X		X

4.10 Deriving elementary problem-solving actions

The possible next step would be deriving elementary problem-solving actions by comparing design solutions with design problems. Out of these elementary problem-solving actions, method fragments are created. In this research, we make a distinction between two phases. An organization utilizes no BRMS or does utilize a BRMS. This is the same for each observed case, and the road to implementing a BRMS is different for each case. When implementing a BRMS in a specific organization, there is no wrong selection of design factors, only the best fit for a specific organizations situation. Nonetheless, it is still possible to create method fragments to support solving design problems when implementing a BRMS. An example of this is as followed: design problem #1 low productivity of elicitation, is proposed to be solved with a certain configuration of characterizing design factors. Design problem #1 could be solved with a combination of the following design factors: #5 Capability autonomy (and the specific level of autonomy), #6 Elicitation source (Data), #7 Relevant set of selected sources, #8 Scenario analysis, and #14 Role: Input. Design factor #5, #6, #7, #8 and #14 result into method fragment #2, #3, #4, #5, and #12. The combination of these design factors and their created method fragments will evolve into a method which could solve design problem #1 as shown in Figure 4.

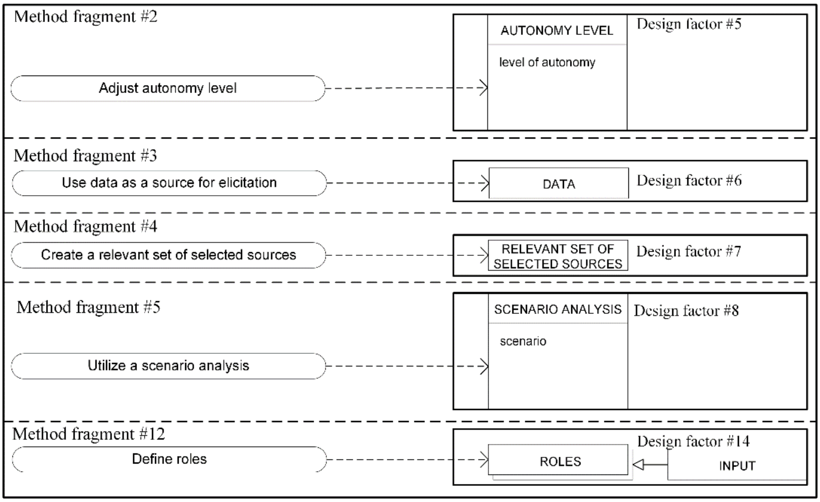

Figure 4: Problem-solving actions

4.11 Deriving method configuration rules

Based on the set of identified design problems and specified method fragments, method configuration rules need to be derived. Basically, the (reusable) method fragments identified in the previous section need to be related to their respective design situations. Configuration rules need to be designed which guides the configuration of solutions to solve specific design problems, as shown in Table 4. Merging the method fragments into one super method is not sufficient for solving the design problems. A certain combination of design problems and design factors requires additional information and attention. Therefore, characterizing design factors related to a certain problem class automatically means that a whole problem class (capability) is implemented. It is not possible to only implement certain parts of a problem class, this is because of the dependencies of the solutions. Therefore, the whole problem class is implemented together with the related non-characterizing design factors and the related characterizing design factors. Certain design situations indicate that BRMS implementation cases exist which deliberately did not have a design factor implemented even if this design factor should solve a specific design problem. The configuration of a design situation depends on design factors which identify the type of implementation. The design factors identifying the types of implementations are focused on the sector in which the organization operates, the number of employees working at this organization, and the focus of the implementation. A specific value of these three design factors gives an indication which design situation fits the organization needs.

Table 4: Method configuration rules

Design problem:	Implement a method consisting of method fragment:
Increase Elicitation productivity	N1, N2, and #1 - #13
Construct library of decisions	N1, N2, N4, N5, #1 - #13, #17 - #21, and #22 - #37
Ensure artifact relationship insight	N2, N6, #1, #2, #6 - #13, #24 - #36, and #38

4.12 BRMS metamodel

The BRMS metamodel is the result of the BRMS analysis tool and the BRMS construction process. It contains all the identified elements in the BRMS construction process together with their specific values and the relations between these elements. Previous sections resulted in the identification of important elements of a BRMS together with their specific values and are included in the BRMS metamodel, as shown in Figure 5. This metamodel shows the relations between the identified elements with each their specific number present in a BRMS implementation. The BRMS metamodel consists of the identified SITUATIONAL FACTORS: EMPLOYEE RANGE, SECTOR, and IMPLEMENTATION FOCUS, the identified PROBLEM CLASSES: ELICITATION, DESIGN, SPECIFICATION, VERIFICATION, VALIDATION, DEPLOYMENT, EXECUTION, MONITORING, and GOVERNANCE, the number of DESIGN PROBLEMS, METHOD FRAGMENTS, DESIGN FACTORS, and DESIGN SITUATIONS, and together with their relationships. In the future, the BRMS metamodel could be used as a reference model for tools focussed on supporting the implementation of a BRMS.

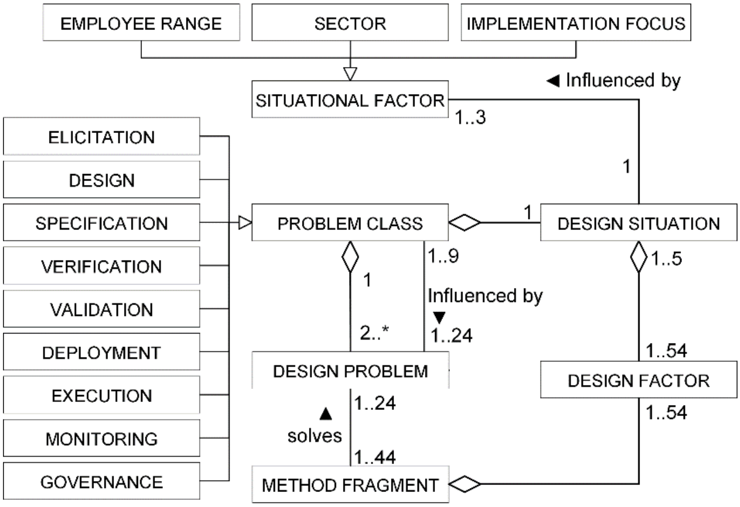

Figure 5: BRMS metamodel

5 Validation

When following the initial situational artefact construction technique, one of the first additions made was regarding the validation part of the results, specific with people from practice. The research fields of BRM and BRMS are lacking standardization and therefore comparison, and validation is needed from practice (Nelson et al., 2008; Zoet, 2014). Semi-structured interviews were conducted with the focus on validating the correctness of the BRMS implementation framework and it's containing elements. The same selection is made as with the validation of the BRMS analysis tool (Leewis, Smit, & Zoet, 2018), which focussed on experts with experience in the BRM and BRMS research field. In total, three experts were interviewed. The first expert is a professor lecturing and performing research in the field of BRM and BRMS. The second expert is a lecturer and PhD candidate with 6 years of practical and research experience in the field of BRM and BRMS. The third expert is a lecturer with 3 years of practical and research experience on BRMS capabilities. An interview protocol was created for the expert interviews to ensure a structured and consistent line of questioning. The major elements (BRMS analysis tool, BRMS construction process, and BRMS metamodel) of the BRMS framework and the sub-elements (Problem Classes, Design Situations, Method Fragments, and Method Configuration Rules) were proposed to the experts. The experts indicated which possible changes should be made and which elements should be included or excluded. Besides stating whether elements are correct or not correct, examples from practice were asked from the experts to support the validity of the BRMS implementation framework.

6 Conclusions

The goal of this research was to develop a framework to support the implementation of a business rules management solution from an organizational perspective. In order to achieve this goal, we explored the business rules management problem space and its neighbouring fields on techniques to create a framework. We selected the situational artefact construction technique of Winter (Winter, 2011a, 2011b) as a guideline for creating our framework. To be able to create a situational artefact, the state of the art of the BRM and BRMS research field needed to be explored. The state of the art literature review results with regards to the BRMS analysis tool (Leewis, Smit, & Zoet, 2018) was used as a reference for this study. The 13 BRMS implementations gathered using the BRMS analysis tool (Leewis, Smit, & Zoet, 2018) were used as input for the BRMS construction process. Analysis of the BRMS implementation data using the BRMS construction process resulted in elements to support the creation of the BRMS implementation framework. The elements identified in the BRMS analysis tool and the BRMS construction process resulted in the BRMS metamodel, as presented in Figure 5. Because of the explorative nature of this study the composition of the BRMS metamodel could change as the maturity of the research field improves.

We believe that creating this framework to support BRMS implementations from an organizational perspective will contribute to the maturity of the BRM and BRMS field, creating a foundation towards other situational artefact research in general and in the

BRM and BRMS field. However, we believe certain aspects of this research are susceptible for discussion. One of the main limitations of this research is that situational artefact construction relies on large samples (60+ BRMS implementation cases at minimum) as input for the data analysis. Our sample consists of only 13 BRMS implementation cases. The initial quantitative PCA was replaced with a qualitative approach due to the fact of the small sample (n=13). Furthermore, the small sample size had an influence on the instantiation of design factors. The number of experts ($n = 3$) used for validation of the BRMS analysis tool and the BRMS implementation framework are identified as a limitation and as a threat to the construct validity, and reliability. Being that the BRMS analysis tool is an important element of creating the BRMS implementation framework. The possibility exists that the experts, if biased, have a higher impact on the validity of the BRMS analysis tool and the BRMS implementation framework when low in numbers.

The BRMS implementation framework is validated by experts in an expert interview. However, to support a real-world BRMS implementation, real-world proof is needed that the framework is valid and correct. Lastly, the exploratory nature of this research concerning the use of situational artefact construction in an immature field. It is still not proven adequately that using this technique is possible in immature fields which could pose threats to validity as well. Based on these points of discussion we argue that future research is needed in the field of situational artefact construction with a focus on immature fields.

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459. http://doi.org/10.1002/wics.101

Aier, S., Riege, C., & Winter, R. (2008). Classification of enterprise architecture scenarios – an exploratory analysis. *Enterprise Modelling and Information Systems Architectures*, *3*(1), 14–23. http://doi.org/10.18417/emisa.3.1.2

Arnott, D., & Pervan, G. (2014). A critical analysis of decision support systems research revisited: The rise of design science. *Journal of Information Technology*, *29*(4), 269–293. http://doi.org/10.1057/jit.2014.16

Baumöl, U. (2005). Strategic agility through situational method construction. In *Proceedings of EURAM 2005 (European Academy of Management)*. Munchen.

Bekkers, W., van de Weerd, I., Brinkkemper, S., & Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study. In *2008 Second International Workshop on Software Product Management* (pp. 43–50).

Boyer, J., & Mili, H. (2011). *Agile business rule development*. *Agile Business Rule Development*. Berlin, Heidelberg: Springer. http://doi.org/10.1007/978-3-642-19041-4

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275–280. http://doi.org/10.1016/0950-5849(95)01059-9

Bucher, T., & Winter, R. (2010). Taxonomy of business process management approaches. In *Handbook on Business Process Management* (pp. 93–114). New York, NY: Springer.

Business Rules Group. (2003). *The Business Rules Manifesto*. Retrieved from http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf

Deza, M. M., & Deza, E. (2016). *Encyclopedia of Distances* (4th ed.). Berlin: Springer Berlin Heidelberg. Retrieved from http://www.springer.com/us/book/9783642002342

Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster Analysis* (5th ed.). London: Wiley. http://doi.org/10.1007/BF00154794

Graham, I. (2007). *Business rules management and service oriented architecture a pattern language* (1st ed.). Hoboken, NJ: John Wiley & Sons.

Hardle, W., & Simar, L. (2007). *Applied multivariate statistical analysis* (2nd ed.). Berlin: Springer-Verlag. http://doi.org/10.1007/978-3-540-72244-1

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *1*(28), 75–105.

Jolliffe, I. T. (2002). *Principal Component Analysis, Second Edition*. *Springer Series in Statistics* (Vol. 98). New York, NY: Springer Berlin Heidelberg. http://doi.org/10.1007/b98835

Karlsson, F., Ågerfalk, P. J., & Hjalmarsson, A. (2001). Method configuration with development tracks and generic project types.

Klesse, M., & Winter, R. (2007). Organizational forms of data warehousing: An explorative analysis. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (p. 11). http://doi.org/10.1109/HICSS.2007.417

Leewis, S., Smit, K., & Zoet, M. (2018). A Tool for Analyzing Business Rules Management Solution Implementations. In *eKNOW 2018 : The Tenth International Conference on Information, Process, and Knowledge Management Performance* (pp. 95–100). Rome.

Leewis, S., Smit, K., Zoet, M., & Berkhout, M. (2018). Solving Problems by Implementing a Business Rules Management System. In *eKNOW 2018 : The Tenth International Conference on Information, Process, and Knowledge Management Performance* (pp. 89–94). Rome.

Morgan, T. (2002). *Business Rules and Information Systems : Aligning IT with Business Goals*. Boston, MA: Addison-Wesley.

Nelson, M. L., Peterson, J., Rariden, R. L., & Sen, R. (2010). Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry. *Information and Management*, *47*(1), 30–41. http://doi.org/10.1016/j.im.2009.09.007

Nelson, M. L., Rariden, R. L., Sen, R., & Texas, A. (2008). A Lifecycle Approach towards Business Rules Management. In *41st Hawaii International Conference on System Sciences* (pp. 1–10).

Ravesteyn, P., & Jansen, S. (2009). A Situational Implementation Method for Business Process Management Systems. In *Proceedings of the Fifteenth Americas Conference on Information Systems (AMCIS)* (pp. 1–9). San Francisco, CA.

Rolland, C., & Prakash, N. (1996). A proposal for context-specific method engineering. In *Method Engineering* (pp. 191–208). Springer.

Rosca, D., & Wild, C. (2002). Towards a flexible deployment of business rules. *Expert Systems with Applications*, *23*(4), 385–394.

Ross, R. G. (2003). *Principles of the business rule approach* (1st ed.). Boston, MA: Addison-Wesley Professional.

Ross, R. G. (2013). What You Need to Know About Business Rules. *Business Rule Concepts*, 23–31.

Smit, K., & Zoet, M. (2016). Management control system for business rules management. *International Journal on Advances in Systems and Measurements (IARIA)*, *9*(3), 210–219.

Smit, K., Zoet, M., & Berkhout, M. (2017). Functional Requirements for Business Rules Management Systems. *Twenty-Third Americas Conference on Information Systems*, 1–10.

Taylor, J. (2011). *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics* (1st ed.). Indianapolis, IN: IBM press.

van de Weerd, I. (2009). *Advancing in software product management: An incremental method engineering approach*. Utrecht: Utrecht University.

van der Aalst, W. M. P. (1998). Three Good reasons for Using a Petri-net-based Workflow

Management System. In *Information and Process Integration in Enterprises: Rethinking Documents* (pp. 161–182).

van Slooten, K., & Hodes, B. (1996). Characterizing IS Development Projects. *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, 26-28 August 1996, Atlanta, USA*, (1992), 29–44. http://doi.org/10.1007/978-0-387-35080-6_3

Winter, R. (2011a). Design Solution Analysis for the Construction of Situational Design Methods. In *Engineering Methods in the Service-Oriented Context* (pp. 19–33). Berlin: Springer.

Winter, R. (2011b). Problem analysis for situational artefact construction in information systems. In *Emerging themes in information systems and organization studies* (pp. 97–113). Berlin: Springer.

Zoet, M. (2014). *Methods and Concepts for Business Rules Management*. Utrecht: Hogeschool Utrecht.

Zoet, M., & Versendaal, J. (2013). Business Rules Management Solutions Problem Space: Situational Factors. In *Pacific Asia Conference on Information Systems 2013 (PACIS)* (p. 247).